

**Applicant:** Gazda et al.  
**Application No.:** 10/648,155

**REMARKS/ARGUMENTS**

Claims 1-3 are currently pending in this application. The Examiner rejected claims 1-3 under 35 U.S.C. §102. The Applicants have canceled claim 2 and 3, and have amended claim 1 to more particularly and distinctly point out the subject matter the Applicants regard as the invention. Applicants have also added new claims 6-9.

All amendments and new claims 6-9 are fully supported in the specification. Applicants submit that no new matter has been added.

**35 U.S.C. §102(b) – Wilkinson et al.**

The Examiner rejected claims 1-3 under 35 U.S.C. §102(b) as being anticipated by Wilkinson et al. (U.S. Publication No. 20020091800).

The Examiner states that Wilkinson discloses an operating system abstraction layer comprising an interface with an operating environment, the operating environment operating independent of underlying operating systems; and wherein the operating environment hides underlying operating systems from its client applications; and an operating system independent module for performing operations that are not related to a target operating system; and an operating system dependent module for performing operations that are related to the target operating system and an interface with the target operating system.

The Wilkinson reference includes drawings that show “boxes” labeled with the names of these components. However, there is no disclosure, teaching or suggestion in the Wilkinson reference that would enable one skilled in the art to create an operating system abstraction layer. Wilkinson only describes the vaguest of concepts relative to an operating system abstraction layer. The most detailed disclosure of an operating system abstraction layer given by Wilkinson is found in paragraph [0060], which recites:

“The operating system (OS) layer includes an abstraction layer which enables the use of a standard operating system 16 and usage of third party components like device drivers, and also provides the flexibility

to exchange these operating systems without affecting the rest of the system.”

Thus the Wilkinson reference only discloses that an OS abstraction layer exists and what it is supposed to do. It teaches nothing about how to create an abstraction layer, nor does it enable one skilled in the art to create an operating system abstraction layer.

On the other hand, the Applicants fully disclose the nature of the operating system abstraction layer including the constructs, modules and API necessary to create and use an operating system abstraction layer, supported by full and very detailed disclosure in the specification. For example, claim 7 discloses the constructs that comprise the OS abstraction layer including a thread, a process, a thread group, a mutex and an event. Claim 9 discloses the OS modules that comprise the abstraction layer, such as a mutex module, an event module, a thread module, a message module, a generic list module, a message queue module, and a memory management module. Accordingly, the Applicants' amended claim 1 and newly added claims 6-9 are patentable over the Wilkinson reference.

**Conclusion**

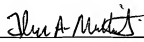
If the Examiner believes that any additional minor formal matters need to be addressed in order to place this application in condition for allowance, or that a telephone interview will help to materially advance the prosecution of this application, the Examiner is invited to contact the undersigned by telephone at the Examiner's convenience.

In view of the foregoing remarks, Applicants respectfully submit that the present application, including claim 1 and claims 6-9, is in condition for allowance and a notice to that effect is respectfully requested.

Reconsideration and entry of this amendment is respectfully requested.

Respectfully submitted,

Gazda et al.

By   
Thomas A. Mattioli  
Registration No. 56,773

Volpe and Koenig, P.C.  
United Plaza, Suite 1600  
30 South 17th Street  
Philadelphia, PA 19103  
Telephone: (215) 568-6400  
Facsimile: (215) 568-6499

TAM/RIS/jmn